COMPARISON OF STATIC AND DYNAMIC LOAD BALANCING IN GRID COMPUTING

Mr. Ramesh Prajapati[1], Mr. Dushyantsinh Rathod[2], Dr. Samrat Khanna[3]
[1,2]Dept. of Computer Science Engineering, Rai University, Center for Research & Development
Saroda, Dholka, India
[3]Dept. of Information Technology, Istar, Sardar Patel centre for science & Tech.
V.Vnagar, India

Abstract: Grid computing consists of a number of heterogeneous systems sharing resources toward a common goal. Management of resource and Workload are two main functions by Grid. Grid computing is to create large and powerful self-managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. Load balancing is used to improve scalability and overall system throughput in Grid computing. Load balancing improves the system performance by dividing the work load effectively among the participating nodes. To effective and efficient load balancing are fundamentally important Load balancing enables in effective technique of resources to improve the overall performance of the system. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both job response time and resource utilization while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or lightly loaded. The purpose of this paper is to Comparison of Static and Dynamic load balancing for the grid based distributed network

Keywords: Grid computing, Load balancing, static load Balancing, dynamic load balancing

I. INTRODUCTION

Grid is a generalized network computing system that is supposed to scale to Internet levels and handle data and computation seamlessly. In Grid computing, individual users can access computers and data, transparently, without having to consider location, operating system, account administration, and other details. Grids tend to be more loosely coupled, heterogeneous, and geographically distributed. Grid computing is the act of sharing tasks over multiple computers. Tasks can range from data storage to complex calculations and can be spread over large geographical distances [1]. Grid technology allows organizations to use numerous computers to solve problems by sharing computing resources. The problems to be solved might involve data processing, network bandwidth, or data storage. With the increased popularity of internet and availability of high performance computers and high speed networks as low cost commodity, it has become possible to use networks of computers as a single unified computing resource. In Grid computing, individual users can access computers and data, transparently, without having to consider location, operating system, account administration, and other details. Grids tend to be more loosely coupled, heterogeneous, and geographically distributed. In Grid computing details are abstracted, and the resources are virtualized. Grid resources are distributed geographically in a large-scale way and resource performance changes quickly over time. Grid computing is enabled by relatively high-performance computers, robust computer networks, grid management software, and the divisibility of difficult scientific problems. In Grid information sharing among processors, resource sharing, shorter response time, high throughput, better price/performance ratio, higher reliability and extensibility [3]. The major research issue in Grid Computing is development of effective techniques for distributing work load among multiple processors. The main goal is to achieve performance goals along with distribution of work load. Load balancing algorithms enables the jobs to move from one processor to another. Load balancing[4] is defined as a process of allocating the total work load to the individual nodes of the distributed systems to improve resource utilization and response time, also avoiding the condition in which some nodes are overloaded while others are under loaded or server failure. In general, as the complexity of distributed systems grows, their load balancing requires more advancement functionality, such as the ability to tolerate faults, install new load balancing algorithms at run time and to create replica to handle busty clients. Distributed systems performance and scalability will be affected due to the lack of above functionalities. A load balancing algorithm should be general and transparent to the applications, also it should provide minimum overhead to the system [5]. The load balancing process is defined in three rules, first one is location rule which gives information about the resource domain to be included in the balancing operations and second rule is distribution rule which is used in case redistribution among nodes is needed. Lastly selection rule which decides whether the load balancing can be performed or not. Section II describes the Load Balancing. In Section III, we talk about Static Load balancing. Dynamic Load Balancing is presented in Section IV. In section V, we present the Comparison between Static and Dynamic load balancing. We present the conclusion in section VI

II. LOAD BALANCING

A main characteristic of Grids [10] is that resources are shared among numerous applications, and therefore, the amount of resources available to any given application highly fluctuates over time. Load balancing is a technique to enhance resources, utilizing parallelism, exploiting throughput improvisation, and to cut response time through an appropriate distribution of the application. To minimize the decision time is one of the objectives for load balancing which has yet not been achieved Load balancing[4] is very effective technique to reduce response time and to improve resources utilization, exploiting through proper distribution of the application assuming homogeneous set of nodes linked with homogeneous and fast networks, various load balancing algorithms were developed. Analyzing the past results and to improve the performance and throughput, efficient algorithms with better scheduling policies [5]. Load balancing is the assignment of work to processors is critical in parallel simulations. It maximizes application performance by keeping processor idle time and interprocessor communication as low as possible. Load balancing algorithms enables the jobs to move from one processor to another. Various studies shown that load balancing among distributed systems improve resource utilization along with system performance. Load balancing is a mechanism[16] to enhance resources, exploiting throughput, utilizing parallelism improvisation, and to cut response time through an appropriate distribution of the applications. To minimize the decision time is one of the objectives for load balancing which has yet not been achieved. Job migration is the only efficient way to guarantee that submitted jobs are completed reliably and efficiently in case of process failure, processor failure, node crash, network failure, system performance degradation, communication delay; addition of new machines dynamically even though a resource failure occurs which changes the distributed environment. Load balancing feature can prove invaluable for handling occasional peak loads of activity in parts of a larger organization. These are important issues in Load Balancing[19]:

- An unexpected peak can be routed to relatively idle machines in the Grid.
- If the Grid is already fully utilized, the lowest priority work being performed on the Grid can be temporarily suspended or even cancelled and performed again later to make room for the higher priority work.

In Grid service [18], a provider may have heterogeneous clusters of resources offering a variety of services to widely distributed user communities. Within such a provision of services, it will be desirable that the clusters will be hosted in a cost effective manner. Hence, an efficient structure of the available resources should be decided upon these clusters. A static load balancing [21], adopted in classical distributed systems, where a single master node controls all resources and decides where incoming jobs should be executed is not efficient for Grid computing. For this purpose, we propose a dynamic load balancing [19] to represent Grid architecture in order to manage workload. In proposed model, we compare static and dynamic load balancing algorithms and each with its advantages and disadvantages.

III. STATIC LOAD BALANCING

Static load-balancing algorithms [21] assume that a priori information about all of the characteristics of the jobs, the computing nodes and the communication network are known. Load-balancing decisions are made at compile time, Transfer decisions are independent of the actual current system state. In Static Load Balancing the performance of the processor is determined at the beginning of the execution and the tasks are assigned to the individual processors by the master processor and are always executed by the same processor to which it is assigned. Then depending upon their performance the work load is distributed in the start by the master processor [4]. The slave processors calculate their allocated work and submit their result to the master. A task is always executed on the processor to which it is assigned that is static load balancing methods are no preemptive. Static load balancing method is to reduce the overall execution time of a concurrent program while minimizing the communication delays. A disadvantage of all static schemes is that the final selection of a host for process allocation is made when the process is created and cannot be changed during process execution to make changes in the system load. There are four types of static load balancing[22]: - Round Robin algorithm, Randomized algorithm, Central Manager Algorithm, and Threshold algorithm. Round Robin Algorithm[5]: it distributes jobs evenly to all processors. Processors perform locally on each process independent of allocation of other processor. Randomized Algorithm [5]: it uses random numbers to choose the processors. The random numbers are generated based on a statistic distribution. Central Manager Algorithm [6]: in this a central process will choose a slave processor and assign a job. The selection is based the least load. Threshold Algorithm [6]: here the processors are assigned immediately upon creation to host two threshold parameter under and upper are used to describe the three main levels, under loaded, medium loaded and over loaded.

The advantages of Static Load Balancing are:

No execution overhead. The methods are very simple. Minimal communication delay.

The disadvantages of Static Load Balancing are:

it is very difficult to estimate a-priori the execution time of various parts of a program. Sometimes there are communication delays that vary in an uncontrollable way for some problems the number of steps to reach a solution is not known in advance No accurate methods to estimate execution time. The process allocation cannot be changed during execution. These methods do not consider data distribution complications.

IV. DYNAMIC LOAD BALANCING

Dynamic load-balancing algorithms [11] attempt to use the run-time state information to make more informative decisions in sharing the system load. It allocate the tasks of a parallel program to workstations Dynamic Load Balancing algorithm makes use of system state information and allocates the jobs to the processor during run time and this can be changed as the circumstances changes [7]. Here continuous monitoring of load on all the processors is done and whenever the load imbalance reaches some predefined level the redistribution of work is done. This causes an extra overhead at execution time. There are different strategies required by Dynamic Load Balancing algorithm. Load distribution decision based on current work load. Mechanism for collecting and managing system state information. Mechanisms to assist each node in deciding which job is eligible for load balancing and assist job transfer from local to remote node. Mechanism on which the destination node is selected.

Based on the above requirements three main strategies are used [10].

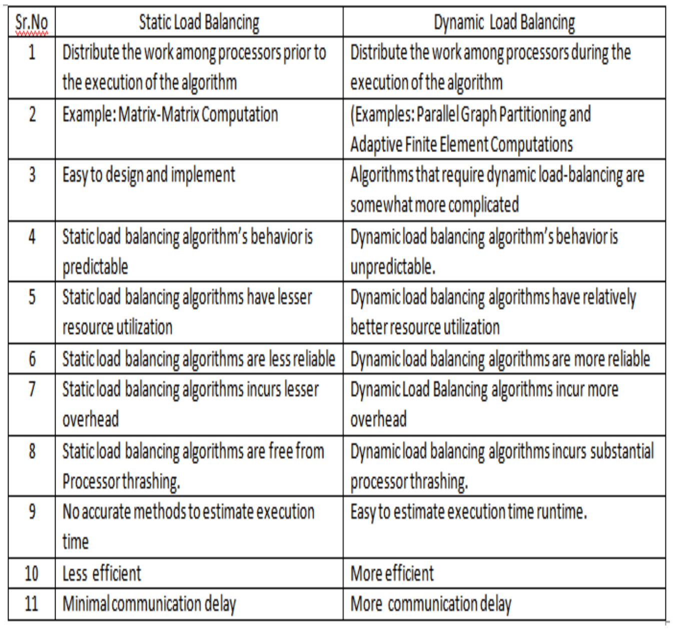
- Information strategy: This is responsible for collecting the information about the nodes.
- Transfer strategy: This select the job for selection of a job for transfer from a local node to remote node.
- Location strategy: This is to select a destination node for transferred task.

There are two types of Dynamic Load Balancing algorithm. Central Queue Algorithm [14]: the principle of dynamic distribution. It stores new activities and unfulfilled request as a cyclic FIFO queue on the main host. Local queue Algorithm [14]: it works on the principle of process migration. Initially the static allocation of the processors is done and process migration is initiated by a host when the load falls under some user defined threshold limit.

The general advantages of Dynamic Load Balancing are:

More efficient compared to Static Load Balancing. Runtime decisions are done. The general disadvantages of of Dynamic Load Balancing are: Centralized algorithm cause bottle neck in the system. More suitable for networks with smaller size it works on the principle of dynamic distribution.

V. COMPARISON OF STATIC & DYNAMIC LOAD BALANCING



Sr.No	Static Load Balancing	Dynamic Load Balancing
1	Distribute the work among processors prior to the execution of the algorithm	Distribute the work among processors during the execution of the algorithm
2	Example: Matrix-Matrix Computation	(Examples: Parallel Graph Partitioning and Adaptive Finite Element Computations
3	Easy to design and implement	Algorithms that require dynamic load-balancing are somewhat more complicated
4	Static load balancing algorithm's behavior is predictable	Dynamic load balancing algorithm's behavior is unpredictable.
5	Static load balancing algorithms have lesser resource utilization	Dynamic load balancing algorithms have relatively better resource utilization
6	Static load balancing algorithms are less reliable	Dynamic load balancing algorithms are more reliable
7	Static load balancing algorithms incurs lesser overhead	Dynamic Load Balancing algorithms incur more overhead
8	Static load balancing algorithms are free from Processor thrashing.	Dynamic load balancing algorithms incurs substantial processor thrashing.
9	No accurate methods to estimate execution time	Easy to estimate execution time runtime.
10	Less efficient	More efficient
11	Minimal communication delay	More communication delay

TABLE I: Comparison of Static and Dynamic Load Balancing

VI. CONCLUSION

This paper presents a survey of load balancing and Static and Dynamic Load Balancing algorithms in grid environment. Grid Computing is Heterogeneous, Dynamic nature and complexity of Grid make load balancing very complex and vulnerable to faults. To maintain entire load of nodes is very hard due to dynamic nature of resources in a Grid environment. Resources submitted to Grid and can be withdrawn from Grid at any moment. Main Goal of load balancing algorithm is to achieve high performance in grid environment by optimal usage of geographically distributed and heterogeneous resources. Here we compare static and dynamic load balancing algorithm and which is best method for load balancing. The above comparison shows that static load balancing algorithms are more stable than dynamic algorithms but due to capability of performing accurate in grid computing, dynamic load balancing is chosen over static load balancing algorithms. This analysis further can also help in designing new load balancing algorithms In future work proposes an efficient Dynamic load balancing algorithm for Grid environment.

RERERENCES

[1] M. Dobber, R. Mei, and G. Koole, "Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment: A Comparison", IEEE Transaction on Parallel and Distributed Systems, Vol. 20, no. 2, pp. 207-218, February 2009.

[2] B. Yagoubi , Department of Computer Science, Faculty of Sciences, University of Oran and Y. Slimani , Department of Computer Science, Faculty of Sciences of Tunis, Task Load Balancing Strategy for Grid Computing

[3] R. Buyya and J. Giddy and H. Stockinger, Economic Models for Resource Management and Scheduling in Grid Computing, in J. of Concurrency and Computation: Practice and Experience, Volume 14, Issue (13-15), Pages (1507-1542), Wiley Press, Dec. 2002

[4] Hisao Kameda, El-Zoghdy Said Fathyy and Inhwan Ryuz Jie Lix, "A Performance Comparison of Dynamic vs.Static Load Balancing Policies in a Mainframe { Personal Computer Network Model", Proceedings of the 39th IEEE Conference on Decision and Control, 2000.

[5] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma,"Performance Analysis of Load Balancing Algorithms",academy of science, engineering and technology, issue 38,February 2008, pp. 269-272

[6] M. Dobber, R. Mei, and G. Koole, "Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment: A Comparison", IEEE Transaction on Parallel and Distributed Systems, Vol. 20, no. 2, pp. 207-218, February 2009.

[7] Abhijit A. Rajguru, S.S. Apte, "A Comparative Performance Analysi of Load Balancing Algorithms in Distributed System using Qualitative Parameters", International Journal of recent

Technology and Engineering, Volume 1, Issue 3, August 2012.

[8] Foster, I., Kesselman, C. and Tuecke, S. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International Journal of High Performance Computing applications, Vol. 15, No. 3, pp. 200-222,2001

[9] Book - "Introduction to Grid Computing (Hardcover - 2009)" (url - http://www.flipkart.com/introduction-grid-computing-magoules-frederic/1420074067-eox3f9kfvb)

[10] Ratnesh Kumar Nath, "Efficient Load Balancing Algorithm in Grid Environment", Thapar University, Patiala, May 2007.

[11] Belabbas Yagoubi and Yahya Slimani, "Dynamic Load Balancing Strategy for Grid Computing", World Academy of Science, Engineering and Technology 19, 2006.

[12] Javier Bustos Jimenez, "Robin Hood: An Active Objects Load Balancing Mechanism for Intranet", Departamento de Ciencias de la Computacion, jbustos@dcc.uchile.cl, Universidad de Chile.

[13] S. Iqbal, Load balancing strategies for parallel architectures, Ph.D. Thesis, Univeristy of Texas at Austin, May 2003

[14] Shahzad Malik, "Dynamic Load Balancing in a Network of Workstations", 95.515F Research Report, November 29, 2000.

[15] Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis and Eduardo Gómez-Sánchez, "Grid Characteristics and Uses: a Grid Definition"

[16] Kai Lu, Riky Subrata and Albert Y. Zomaya, Networks & Systems Lab, School of Information Technologies, University of Sydney "An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems Considering Desirability of Grid Sites".

[17] http://images.google.co.in/images.

[18] Frederic Magoules, Jie pan, Kiat-An Tan, Abhinit Kumar, "Introduction to Grid Computing", CRC Press, A chapman and hall Book.

[19] An Effective Dynamic Load Balancing Algorithm for Grid System" International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 8- August 2013.

[20] Abhijit A. Rajguru, S.S. Apte "AComparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters" International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-1, Issue-3, August 2012.

[21] Hendra Rahmawan, Yudi Satria Gondokaryono, "The Simulation of Static Load Balancing Algorithms", 2009 International Conference on Electrical Engineering and Informatics, Malaysia.

[22] Toufik Taibi, Abdelouahab Abid, Engku Fariez Engku Azahan ,"A Comparison of Dynamic load Balancing Algorithms", J.J Appl.Science, Volume 9, Issue 2, 2007.